

Scheduling in Wireless Networks under Uncertainties: A Greedy Primal-Dual Approach

Qiao Li

qiaoli@cmu.edu

Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

Rohit Negi

negi@ece.cmu.edu

Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

Abstract—This paper proposes a dynamic primal-dual type algorithm to solve the optimal scheduling problem in wireless networks subject to uncertain parameters, which are generated by stochastic network processes such as random packet arrivals, channel fading, and node mobilities. The algorithm is a generalization of the well-known max-weight scheduling algorithm proposed by Tassiulas *et al.*, where only queue length information is used for computing the schedules when the arrival rates are uncertain. Using the technique of fluid limits, sample path convergence of the algorithm to an arbitrarily close to optimal solution is proved, under the assumption that the Strong Law of Large Numbers (SLLN) applies to the random processes which generate the uncertain parameters. The performance of the algorithm is further verified by simulation results. The method may potentially be applied to other applications where dynamic algorithms for convex problems with uncertain parameters are needed.

I. INTRODUCTION

Scheduling in wireless networks involves efficiently allocating network resources among competing network users *in the presence of uncertainties*. These uncertainties may be either due to unexpected events, such as link failures, or due to intricate cross-layer interactions in wireless networks. For example, the packet arrival rates may be unknown (e.g., [1], [2]), which depend on upper layer dynamics such as routing and congestion control protocols. As another example, the wireless channel statistics may be also unknown (e.g., [2]), since they depend on complex network events such as channel fading, power control and node mobilities.

In the presence of such uncertain parameters, it may no longer be optimal to use the static allocation approach (e.g., [3]), which produces periodic schedules by solving a static underlying convex optimization problem (which usually has exponential size) with estimated uncertain parameters. In particular, if these uncertain parameters are slowly converging, or time varying, the estimated parameters may fail to track the changes in their true value, which often leads to suboptimal schedules. Further, it may be impractical to estimate the uncertain parameters for large wireless networks, as the number of the parameters may grow fast (e.g., exponentially) with the size of the network. For a simple illustration, consider a wireless network with n links, such that each link randomly switches on or off after certain random number of time slots. In such case, a complete specification of the network topology

probabilities may require as large as $2^n - 1$ parameters, which quickly becomes impossible to estimate as n grows.

On the other hand, online algorithms (such as [1], [2], [4], [6], [7]) are more robust to the changes to the uncertain parameters (such as arrival rates), since they use queue length (which can also be interpreted as prices [9], [10]) information for scheduling in each time slot. For example, it has been shown that [4] such algorithms can achieve network stability even if the “instantaneous rates” of the traffic vary arbitrarily inside the network capacity region. Further, compared to the estimation based approach, these online algorithms may be more scalable to the network size, in the sense that the dimension of the queue length vector corresponds to the number of constraints (such as rate constraint for each link), which usually grows slowly, whereas the number of uncertain parameters can grow very fast (e.g., exponentially).

In this paper we solve a general class of optimal wireless network scheduling problems with uncertain parameters, whose underlying static problem is described by the convex optimization problem OPT in Section II. Essentially, we require that the structures of the convex objective functions and convex constraint functions are known, except the values of the uncertain parameters. These parameters will be generated by certain stochastic processes and observed by the network gradually over time slots. We propose a greedy primal-dual dynamic algorithm (Algorithm 1 in Section III) to achieve the optimal scheduling asymptotically. Using the novel technique of fluid limits [12], optimality can be guaranteed under the assumption that all the network processes generating the uncertain parameters satisfy SLLN (see details in Section II). Note that this assumption is quite mild, since we can guarantee optimality *as long as these processes converge, no matter how slowly the convergence happens*. Thus, intuitively, our algorithm can automatically track the convergence of these processes and correct the mistakes which are made within any finite time history.

Our algorithm is a generalization of the well-known max-weight algorithm [1], which was shown to be throughput optimal for i.i.d arrival processes. Our algorithm is related to, but different from the utility-optimal scheduling algorithm by Neely [2], which achieves the optimal scheduling by cleverly transforming the problem into optimizing the time-average of

the utility (and constraint) functions, to which a dual-type algorithm applies. Our algorithm is also different from the primal-dual algorithm by Neely [5] since we use a scaled queue length (by $1/t$) in the scheduling, which corresponds to the approximated gradient. Stolyar [6] also proposed a primal-dual type scheduling algorithm, and proved its optimality using a fluid limit obtained from a different scaling. Since the fluid scaling in [6] is taken over different systems, it is hard to relate the optimality in the fluid limits to the one in the original system. Finally, our algorithm can be used as a MAC layer solution for the general framework of cross-layer optimization problem (e.g., [4], [7], [8], [10]) for wireless networks.

The organization of the following sections is as follows: In Section II we describe the queueing network model as well as the optimization problem OPT, and in Section III we describe the scheduling algorithm. Section IV proves the optimality of the algorithm, Section V illustrate the algorithm performance in simulation, and finally Section VI concludes this paper.

II. NETWORK MODEL AND PROBLEM FORMULATION

In this section we describe the queueing network and propose the optimization problem OPT. We first introduce the queueing network model.

A. Queueing Network

We consider the scheduling problem at the medium access (MAC) layer of a multi-hop wireless network, where the network is modeled as a set of n links. We assume a time-slotted network model, and in each time slot t , the network is in one of the following states: $\mathcal{M} \triangleq \{1, 2, \dots, M\}$. The network state can be used to model network topology, channel fading, and user mobility, etc. We further assume that these network states can be measured by the user nodes¹, which are assumed to be equipped with channel monitoring devices. We associate each network state $m \in \mathcal{M}$ with a finite set of resource allocation modes $\Xi^{(m)} = \{\xi_1^{(m)}, \xi_2^{(m)}, \dots\}$, where each mode $\xi_k^{(m)} \in \Xi^{(m)}$ corresponds to a configuration of network resource allocation, such as carrier and frequency selection in OFDM systems, spreading codes choice in CDMA systems and time slots assignment in TDMA systems.

Denote \mathbf{y} as the uncertain parameters, which are generated by the stochastic process $\mathbf{Y}(t)$, which is a cumulative vector process whose time average converges to \mathbf{y} . Specifically, the assumptions on $\mathbf{Y}(t)$ are: 1) it is subject to SLLN, i.e., with probability 1 (w.p.1),

$$\lim_{t \rightarrow \infty} \mathbf{Y}(t)/t = \mathbf{y} \quad (1)$$

and 2) it has uniformly bounded increment in each time slot:

$$\|\mathbf{Y}(t) - \mathbf{Y}(t-1)\| \leq Y_{\max}, \forall t > 0 \quad (2)$$

¹Note that although the network states are global, they often allow (approximate) decompositions (e.g., [2]) according to either the geographic structure or channel orthogonality, in which case one only needs to measure local network states.

where $Y_{\max} > 0$ is a finite constant. For specific examples, consider the cumulative network state process $M_m(t)$:

$$S_m(t) = \sum_{\tau=1}^t \mathbb{1}_{\{m(\tau)=m\}} \quad (3)$$

where $m(t)$ is the network state at time slot t , and $\mathbb{1}_{\{\cdot\}}$ is the indicator function, i.e., $\mathbb{1}_{\{\text{true}\}} = 1$ and $\mathbb{1}_{\{\text{false}\}} = 0$. Thus, SLLN implies that w.p.1,

$$\lim_{t \rightarrow \infty} \frac{1}{t} S_m(t) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=1}^t \mathbb{1}_{\{m(\tau)=m\}} = \pi^{(m)}, \forall m \in \mathcal{M}$$

As another example, consider the external packet arrival process $\mathbf{A}(t)$, which is a $n \times 1$ vector representing the cumulative external packet arrivals during the first t time slots. Similarly, SLLN implies that w.p.1,

$$\lim_{t \rightarrow \infty} \mathbf{A}(t)/t = \mathbf{a}, \text{ w.p.1} \quad (4)$$

Further, we require that the maximum packet arrivals in any time slot are uniformly bounded:

$$\|\mathbf{A}(t) - \mathbf{A}(t-1)\| \leq A_{\max}, \forall t > 0 \quad (5)$$

where $A_{\max} > 0$ is a finite constant.

We finally describe the queueing system model. The queuing dynamics of the network is modeled as follows:

$$\mathbf{Q}(t) = \mathbf{Q}(0) + \mathbf{A}(t) - R(m(t))\mathbf{D}(t) \quad (6)$$

where $\mathbf{Q}(t)$ is the queue length vector at time slot t , and $R(m(t))$ is the $n \times n$ routing matrix, such that $R_{ii}(m(t)) = 1$, and $R_{ij}(m(t)) = -1$ only if link i serves as the next hop for link j at time slot t , as specified by certain routing protocols, otherwise $R_{ij}(m(t)) = 0$. Note that the routing matrix $R(m(t))$ is a function of the network state, and therefore SLLN implies

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=1}^t R(m(\tau)) = \sum_{m \in \mathcal{M}} \pi^{(m)} R(m) \quad (7)$$

$\mathbf{D}(t)$ is a $n \times 1$ vector representing the cumulative packet departures during the first t time slots, which are determined by the resource allocation modes as specified by the scheduler in each time slot. Specifically, at each time slot t with network state m , if the scheduler chooses a resource allocation mode $\xi_k^{(m)}$, there is an associated departure vector $\mathbf{G}_k^{(m)}$, whose each entry $G_{ki}^{(m)}$ corresponds to the number of packets transmitted successfully by link i . Note that the choice of resource allocation mode $\xi_k^{(m)}$ is subject to the constraint that $\mathbf{Q}(t) - \mathbf{G}_k^{(m)} \succeq \mathbf{0}$, so that the queue lengths never become negative. Note that this constraint can be easily satisfied in general systems. For example, if the allocation mode $\xi_k^{(m)}$ corresponds to independent sets of the interference graph (see, for example, [7]), one can simply transmit the subset of links with nonempty queues, which are still independent. In a compact form, we can express the departure process as

$$\mathbf{D}(t) = \sum_{m \in \mathcal{M}} G^{(m)} \mathbf{T}^{(m)}(t), \quad (8)$$

where $G^{(m)}$ is a matrix whose columns are $\mathbf{G}_k^{(m)}$, and $\mathbf{T}^{(m)}(t)$ is a vector whose each entry $T_k^{(m)}(t)$ corresponds to the number of time slots that resource allocation mode $\xi_k^{(m)}$ is chosen during the first t time slots.

A basic requirement on the scheduler is that it should achieve rate stability [12], i.e.,

$$\lim_{t \rightarrow \infty} \mathbf{D}(t)/t = \mathbf{a} \quad (9)$$

so that the departure rate of each link is equal to the arrival rate, as required by the underlying static optimization problem OPT , which we formulate in the next subsection.

B. Optimization Problem

In this section we introduce the optimization problem OPT , which is implicitly solved by the optimal schedulers. The problem OPT is as follows OPT :

$$\text{OPT: } \min_{\mathbf{x}} \quad f(\mathbf{x}; \mathbf{y}) \quad (10)$$

$$\text{s.t.} \quad \mathbf{h}(\mathbf{x}; \mathbf{y}) \preceq \mathbf{0} \quad (11)$$

$$\mathbf{x}^{(m)} \succeq 0, \quad \mathbf{1}^T \mathbf{x}^{(m)} = 1, \quad \forall m \in \mathcal{M} \quad (12)$$

In the above formulation, $\mathbf{x}^{(m)}$ as a resource allocation vector when the network state is m . That is, each entry $x_k^{(m)}$ is the asymptotic time fraction (assuming the limit exists for now) that resource allocation mode $\xi_k^{(m)}$ is chosen, during the time slots where the network state is m . Thus, $\mathbf{x}^{(m)}$ is subject to the simplex constraint (12). $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(M)})$ is a big vector representing the total resource allocation vector as specified by the scheduler. $f(\mathbf{x}; \mathbf{y})$ is a general convex cost function of variable \mathbf{x} , and $\mathbf{h}(\mathbf{x}; \mathbf{y})$ is a vector of general convex constraint functions of variable \mathbf{x} . The additional parameter \mathbf{y} represents the uncertain parameters, which is valid under the assumption that the corresponding processes are subject to SLLN. Finally, we assume that both $f(\mathbf{x}; \mathbf{y})$ and $\mathbf{h}(\mathbf{x}; \mathbf{y})$ are continuously differentiable as functions of variables (\mathbf{x}, \mathbf{y}) .

The formulation of OPT is quite general, which can be used to model various applications in the literature. For example, if we want to minimize the total transmission power, we can choose $\mathbf{y} = (\boldsymbol{\pi}, \mathbf{p})$, and choose the cost function as follows

$$f(\mathbf{x}; \mathbf{y}) = \sum_{m \in \mathcal{M}} \pi^{(m)} \mathbf{p}^{(m)T} \mathbf{x}^{(m)} \quad (13)$$

where $\pi^{(m)}$ is the time fraction that the network state is m , and $\mathbf{p}^{(m)}$ is a power vector where each entry $p_k^{(m)}$ corresponds to the power consumption when resource allocation mode $\xi_k^{(m)}$ is chosen at network state m . Thus, the cost function in (13) can be interpreted as the average power consumption by the scheduler. Note that we can also encode the power constraint into $\mathbf{h}(\mathbf{x}, \mathbf{y})$ by choosing $\mathbf{y} = (\boldsymbol{\pi}, \mathbf{p})$ and then choosing

$$\mathbf{h}(\mathbf{x}; \mathbf{y}) = \sum_{m \in \mathcal{M}} \pi^{(m)} P^{(m)} \mathbf{x}^{(m)} - \mathbf{p} \quad (14)$$

where $P^{(m)}$ is a diagonal matrix where each diagonal entry $P_{kk}^{(m)}$ corresponds to the power consumption when resource

allocation mode $\xi_k^{(m)}$ is chosen when the network state is m , and \mathbf{p} is the power constraint vector. In this case, (14) is equivalent to requiring a constraint of \mathbf{p} on the average transmission power. In order to encode the network stability constraint, we can choose $\mathbf{y} = (\boldsymbol{\pi}, \mathbf{a})$ and then choose

$$\mathbf{h}(\mathbf{x}; \mathbf{y}) = \mathbf{a} - \sum_{m \in \mathcal{M}} \pi^{(m)} R(m) G^{(m)} \mathbf{x}^{(m)} \quad (15)$$

Thus, (15) requires that the average external and internal arrivals should be less than the average departures, in which case the network is rate stable.

III. SCHEDULING ALGORITHM

In this section we will describe the algorithm to solve OPT . As a standard approach in solving constrained convex optimization problems [11], we transform OPT into another static “penalized” problem, PEN , to which our scheduling algorithm can directly apply. Based on this, we then introduce the scheduling algorithm which solves PEN and, therefore, also solves OPT .

A. Transformed Problem

Assuming that OPT is strictly feasible, we first change the constraints in (11) as follows

$$\mathbf{h}(\mathbf{x}; \mathbf{y}) + \mathbf{z} = \mathbf{0}, \quad \epsilon \mathbf{1} \preceq \mathbf{z} \preceq z_{\max} \mathbf{1} \quad (16)$$

where $\epsilon > 0$ is a small scalar, and $z_{\max} > 0$ is a sufficiently large constant such that the inequality and equality constraints are equivalent. Denote f_ϵ^* as the optimal cost when the constraint is changed to (16). Thus, the optimal value of OPT is f_0^* . We have the following sensitivity lemma stating that f_ϵ^* is a good approximation of f_0^* with sufficiently small ϵ .

Lemma 1 ([11]): Denote $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_\epsilon$ as two Lagrangian multipliers for f_0^* and f_ϵ^* , respectively. We have

$$|f_0^* - f_\epsilon^*| \leq \epsilon \max(\|\boldsymbol{\lambda}_\epsilon\|_1, \|\boldsymbol{\lambda}_0\|_1) \quad (17)$$

We next define the transformed problem as follows:

$$\begin{aligned} \text{PEN: } \min_{\mathbf{x}, \mathbf{z}} \quad & g(\mathbf{x}, \mathbf{z}; \mathbf{y}) = f(\mathbf{x}; \mathbf{y}) + \beta p(\mathbf{x}, \mathbf{z}; \mathbf{y}) \\ \text{s.t.} \quad & \epsilon \mathbf{1} \preceq \mathbf{z} \preceq z_{\max} \mathbf{1} \\ & \mathbf{x}^{(m)} \succeq 0, \quad \mathbf{1}^T \mathbf{x}^{(m)} = 1, \quad \forall m \in \mathcal{M} \end{aligned}$$

where β is a large constant to control approximation accuracy, and $p(\mathbf{x}, \mathbf{z}; \mathbf{y})$ corresponds to the penalty term, which corresponds to various standard penalty functions [11], e.g.,

$$p(\mathbf{x}, \mathbf{z}; \mathbf{y}) = \frac{1}{\alpha} \|\mathbf{h}(\mathbf{x}; \mathbf{y}) + \mathbf{z}\|^\alpha \quad (18)$$

for $\alpha > 1$. In particular, the standard Lyapunov drift analysis (e.g., [1], [2], [4], [7]) corresponds to the case $\alpha = 2$.

Denote $(\mathbf{x}_p^*, \mathbf{z}_p^*)$ as a solution of PEN . We have the following result holds:

$$\text{Lemma 2: } f(\mathbf{x}_p^*; \mathbf{y}) \leq f_\epsilon^*$$

Proof: Denote $(\mathbf{x}_p^*, \mathbf{z}_p^*)$ as a solution of OPT with constraint in (11) replaced by (16). We have

$$\begin{aligned} f(\mathbf{x}_p^*; \mathbf{y}) &\leq f(\mathbf{x}_p^*; \mathbf{y}) + \beta p(\mathbf{x}_p^*, \mathbf{z}_p^*; \mathbf{y}) \\ &\stackrel{(a)}{\leq} f(\mathbf{x}_o^*; \mathbf{y}) + \beta p(\mathbf{x}_o^*, \mathbf{z}_o^*; \mathbf{y}) \\ &\stackrel{(b)}{=} f_\epsilon^* \end{aligned}$$

where (a) is because $(\mathbf{x}_p^*, \mathbf{z}_p^*)$ solves PEN , and (b) is because $(\mathbf{x}_o^*, \mathbf{z}_o^*)$ satisfy the constraint (16). Thus, the claim holds. ■

In the following we will focus on solving PEN , since Lemma 1 and Lemma 2 guarantee that PEN achieve an objective function value which is arbitrarily close to the optimal in OPT . We next describe the scheduling algorithm.

B. Algorithm Description

The problems OPT and PEN are static. On the other hand, the network is dynamic, and must be described by time series. Therefore, before describing the algorithm, we need to define dynamic counterparts of the static variables \mathbf{x}, \mathbf{y} and \mathbf{z} . We first define empirical resource allocation variable

$$\mathbf{x}^{(m)}(t) = \frac{\mathbf{T}^{(m)}(t)}{\mathbf{1}^T \mathbf{T}^{(m)}(t)}, \quad \forall m \in \mathcal{M} \quad (19)$$

i.e., each entry $x_k^{(m)}(t)$ corresponds to the time fraction that resource allocation mode $\xi_k^{(m)}$ is chosen during the first t time slots, when the network state is m . Note that we have

$$\mathbf{x}^{(m)}(t) \succeq 0, \quad \mathbf{1}^T \mathbf{x}^{(m)}(t) = 1, \quad \forall m \in \mathcal{M} \quad (20)$$

Thus, $\mathbf{x}^{(m)}(t)$ can be interpreted as the empirical value of $\mathbf{x}^{(m)}$ which is defined in PEN . Similarly, we denote the empirical value of the uncertain parameter \mathbf{y} as

$$\mathbf{y}(t) = \mathbf{Y}(t)/t \quad (21)$$

i.e., $\mathbf{y}(t)$ is formed by directly taking the average of the process $\mathbf{Y}(t)$. Further, define the empirical value of \mathbf{z} as

$$\mathbf{z}(t) = \mathbf{Z}(t)/t \quad (22)$$

where the cumulative process $\mathbf{Z}(t)$ is defined by

$$\mathbf{Z}(t) = \sum_{\tau=1}^t \mathbf{u}(\tau) \quad (23)$$

and $\mathbf{u}(\tau)$ is computed by the scheduler in Algorithm 1.

Finally, we introduce some notations. Denote ∇_m and ∇_z as the gradient operator with respect to variables $\mathbf{x}^{(m)}$ and \mathbf{z} , respectively. Further, with an abuse of notation, we use the following abbreviated notations:

$$\begin{aligned} f(t) &\triangleq f(\mathbf{x}(t); \mathbf{y}(t)) \\ p(t) &\triangleq p(\mathbf{x}(t), \mathbf{z}(t); \mathbf{y}(t)) \\ g(t) &\triangleq g(\mathbf{x}(t), \mathbf{z}(t); \mathbf{y}(t)) = f(t) + \beta p(t) \end{aligned}$$

The algorithm is described as in Algorithm 1. Essentially, the algorithm updates the variables $\mathbf{x}(t)$ and $\mathbf{z}(t)$ by computing descent directions $\mathbf{v}^{(m)}(t)$ and $\mathbf{u}(t)$ in **Step 1** and **Step 2**, respectively, where $\mathbf{v}^{(m)}(t)$ is an all-zero vector except an one

Algorithm 1 Optimal Scheduling

Step 1. At each time slot t with network state m , choose allocation mode $\xi_k^{(m)}$, where

$$k \in \arg \min_j (\nabla_m f(t) + \beta \nabla_m p(t))_j \quad (24)$$

Step 2. Choose variable $\mathbf{u}(t)$ such that

$$u_i(t) = \begin{cases} \epsilon & \text{if } (\nabla_z p(t))_i \geq 0 \\ z_{\max} & \text{else} \end{cases} \quad (25)$$

and update variables $\mathbf{x}(t)$, $\mathbf{y}(t)$ and $\mathbf{z}(t)$ accordingly.

at the k -th entry. Further note that constraint $\mathbf{Q}(t) \succeq \mathbf{G}_k^{(m)}$ can be satisfied implicitly with regular cost and penalty functions, i.e., assuming the cost for transmitting a set of links is always no smaller than that of transmitting any of its subsets.

From the definition of $\mathbf{x}(t)$ and $\mathbf{z}(t)$, these processes are naturally updated as follows

$$\begin{aligned} \mathbf{x}^{(l)}(t) &= \mathbf{x}^{(l)}(t-1), \quad l \neq m \\ \mathbf{x}^{(m)}(t) &= \mathbf{x}^{(m)}(t-1) + \frac{1}{\mathbf{1}^T \mathbf{T}^{(m)}(t)} (\mathbf{v}^{(m)}(t) - \mathbf{x}^{(m)}(t-1)) \\ \mathbf{z}(t) &= \mathbf{z}(t-1) + \frac{1}{t} (\mathbf{u}(t) - \mathbf{z}(t-1)) \end{aligned}$$

Thus, Algorithm 1 can be viewed as a stochastic gradient algorithm for PEN , where the randomness comes from the time varying functions $f(t)$ and $p(t)$, which are subject to the changes in uncertain parameters $\mathbf{y}(t)$.

The optimization of (24) requires tracking the variables $\mathbf{x}(t)$ and $\mathbf{y}(t)$, in general. However, in applications the structure of the cost function $f(\mathbf{x}; \mathbf{y})$ and penalty function $p(\mathbf{x}; \mathbf{y})$ often allows a much simpler computation. For example, in the important case of optimal power scheduling, where the cost function is formulated as (13) and the constraint is as (15) with the typical value $\alpha = 2$, we have

$$\nabla_m f(t) + \beta \nabla_m p(t) = \pi^{(m)}(t) (\mathbf{p}^{(m)} + \frac{\beta}{t} (R(m) \mathbf{G}^{(m)})^T \mathbf{Q}(t))$$

where $\pi^{(m)}(t) = \frac{\mathbf{1}^T \mathbf{T}^{(m)}(t)}{\sum_{l \in \mathcal{M}} \mathbf{1}^T \mathbf{T}^{(l)}(t)}$ is the empirical time fraction of network state m . Thus, the optimization in (24) essentially only requires the queue length information (note that $\pi_m(t)$ becomes an irrelevant scaling factor in the optimization). In particular, if we are only interested in the rate stability, i.e., setting the objective function as $f(\mathbf{x}; \mathbf{y}) = 0$, the optimization in (24) is equivalent to

$$k \in \arg \min_j (G^{(m)T} (R(m)^T \mathbf{Q}(t)))_j \quad (26)$$

which is the same as the max-weight back-pressure algorithm proposed by [1]. We finally conclude this section by the following lemma, which formally shows, essentially, the descent property of Algorithm 1.

Lemma 3: The following properties hold for Algorithm 1:

1) $\mathbf{v}^{(m)}(t)$ solves the following problem

$$\begin{aligned} \text{GRAD-X: } & \min_{\mathbf{v}^{(m)}} \quad \nabla_m g(t)^T \mathbf{v}^{(m)} \\ \text{s.t.} \quad & \mathbf{v}^{(m)} \succeq 0, \quad \mathbf{1}^T \mathbf{v}^{(m)} = 1 \end{aligned} \quad (27)$$

2) $\mathbf{u}(t)$ solves the following problem

$$\begin{aligned} \text{GRAD-Z: } & \min_{\mathbf{u}} \quad \nabla_z g(t)^T \mathbf{u} \\ \text{s.t.} \quad & \epsilon \mathbf{1} \preceq \mathbf{u} \preceq z_{\max} \mathbf{1} \end{aligned} \quad (28)$$

Thus, the variables (\mathbf{v}, \mathbf{u}) computed by Algorithm 1 can be interpreted as the points in the feasible region of **PEN** which achieves the minimum inner product with the corresponding (stochastic) gradients.

Proof: For 1), note that **GRAD-X** is a Linear Programming (LP) problem over a simplex, and therefore the solution can be obtained at a vertex [11] with the minimum directional derivative. For 2), note that **GRAD-Z** is an LP over a hypercube, and therefore the solution is obtained at the boundary points. Thus the claim follows by noting that $\nabla_z g(t) = \nabla_z p(t)$, since $f(\cdot)$ is not a function of z . ■

IV. OPTIMALITY PROOF

In this section we will prove the optimality of Algorithm 1. There are two issues to consider: 1) We need to show that Algorithm 1 achieves the optimality of **OPT** asymptotically, and 2) We need to show that Algorithm 1 is feasible for **OPT**, i.e., constraint (11) can not be violated. We first briefly introduce fluid limits, which serves as the key technique for the optimality proof.

A. Fluid Limits

We extend the domain of all processes to continuous time by linear interpolation, and define the fluid scaling of a function $l(t)$ as

$$l^r(t) = l(rt)/r \quad (29)$$

where l can be functions T, Y and Z . It can be shown that these scaled functions are uniformly Lipschitz-continuous. Thus, according to the Arzela-Ascoli Theorem [13], any sequence of functions which is indexed by $\{r_n\}_{n=1}^\infty$, i.e., $(T^{r_n}, Y^{r_n}, Z^{r_n})$, contains a subsequence $\{r_{n_k}\}_{k=1}^\infty$ which converges uniformly on compact sets to a set of absolutely continuous functions (and, therefore, differentiable almost everywhere [13]) $(\bar{T}, \bar{Y}, \bar{Z})$. Define any such limit as a fluid limit. (Note that fluid limits are denoted by a bar.) We next state some properties of the fluid limits.

Lemma 4: The processes in any fluid limit satisfies the following: For any $t > 0$, we have w.p.1,

$$\epsilon \mathbf{1} \preceq \bar{Z}(t)/t \preceq z_{\max} \mathbf{1} \quad (30)$$

and the following properties hold w.p.1: For all $t \geq 0$

$$\bar{Y}(t) = yt \quad (31)$$

$$\mathbf{1}^T \bar{\mathbf{T}}^{(m)}(t) = \pi^{(m)} t \quad \forall m \in \mathcal{M} \quad (32)$$

Proof: (30) follows from Algorithm 1, where each $u_i(t)$ is chosen between ϵ and z_{\max} . (31) and (32) follows directly from the (functional) SLLN. ■

We next define the resource allocation variables and auxiliary variables in fluid limit as follows (one can compare with (19) and (22) for similarities)

$$\bar{x}^{(m)}(t) = \frac{1}{\pi^{(m)} t} \bar{\mathbf{T}}^{(m)}(t) \quad (33)$$

$$\bar{z}(t) = \bar{Z}(t)/t \quad (34)$$

Similarly, define the following variables as the counter parts of $\mathbf{v}^{(m)}(t)$ and $\mathbf{u}(t)$ in Algorithm 1:

$$\bar{v}^{(m)}(t) = \dot{\bar{\mathbf{T}}}^{(m)}(t)/\pi^{(m)} \quad (35)$$

$$\bar{u}(t) = \dot{\bar{Z}}(t) \quad (36)$$

We have the following lemma holds, which states that both $(\bar{x}(t), \bar{z}(t))$ and $(\bar{v}(t), \bar{u}(t))$ are feasible points for **PEN**.

Lemma 5: For any fluid limit and $t > 0$, we have

1) $(\bar{x}(t), \bar{z}(t))$ is feasible for **PEN**:

$$\epsilon \mathbf{1} \preceq \bar{z}(t) \preceq z_{\max} \mathbf{1} \quad (37)$$

$$\mathbf{1}^T \bar{x}^{(m)}(t) = 1, \quad \bar{x}^{(m)}(t) \succeq \mathbf{0}, \quad \forall m \in \mathcal{M} \quad (38)$$

2) $(\bar{v}(t), \bar{u}(t))$ is also feasible for **PEN**:

$$\epsilon \mathbf{1} \preceq \bar{u}(t) \preceq z_{\max} \mathbf{1} \quad (39)$$

$$\mathbf{1}^T \bar{v}^{(m)}(t) = 1, \quad \bar{v}^{(m)}(t) \succeq \mathbf{0}, \quad \forall m \in \mathcal{M} \quad (40)$$

3) The derivatives of $\bar{x}^{(m)}(t)$ and $\bar{z}(t)$ are

$$\dot{\bar{x}}^{(m)}(t) = (\bar{v}^{(m)}(t) - \bar{x}^{(m)}(t))/t \quad (41)$$

$$\dot{\bar{z}}(t) = (\bar{u}(t) - \bar{z}(t))/t \quad (42)$$

Proof: For 1), (37) follows from applying (30) to the definitions of $\bar{z}(t)$, and (38) follows from applying (32) to the definition of $\bar{x}^{(m)}(t)$. Similarly we can prove 2), by noting that $\epsilon \tau \mathbf{1} \preceq \bar{Z}(t + \tau) - \bar{Z}(t) \preceq z_{\max} \tau \mathbf{1}$ for any $t \geq 0$ and $\tau > 0$. 3) follows from direct calculation. ■

We are now ready to prove the optimality of Algorithm 1.

B. Optimality Proof

For the ease of presentation, we use

$$\bar{g}(t) \triangleq g(\bar{x}(t), \bar{z}(t); \bar{y}(t)) \quad (43)$$

as a short-hand notation (note that they are the functions in fluid limits), with an abuse of notation. We next establish the following key technical lemma, which, essentially, extends the optimality property in Lemma 3 to the fluid limits.

Lemma 6: Let a fluid limit $(\bar{T}, \bar{Y}, \bar{Z})$ and $m \in \mathcal{M}, t > 0$ be given. The following properties hold:

1) $\bar{v}^{(m)}(t)$ solves the following problem

$$\text{GRAD-XBAR: } \min_{\bar{v}^{(m)}} \quad \nabla_m \bar{g}(t)^T \bar{v}^{(m)} \quad (44)$$

$$\text{s.t.} \quad \bar{v}^{(m)} \succeq 0, \quad \mathbf{1}^T \bar{v}^{(m)} = 1$$

2) $\bar{\mathbf{u}}(t)$ solves the following problem

$$\begin{aligned} \text{GRAD-ZBAR: } & \min_{\bar{\mathbf{u}}} \quad \nabla_{\mathbf{z}} \bar{g}(t)^T \bar{\mathbf{u}} \\ & \text{s.t.} \quad \epsilon \mathbf{1} \preceq \bar{\mathbf{u}} \preceq z_{\max} \mathbf{1} \end{aligned} \quad (45)$$

Thus, the optimality in Lemma 3 still holds in fluid limits.

We first outline the proof. For 1), since GRAD-XBAR is an LP over a simplex, the optimum must correspond to the vertices with the smallest gradient. Thus, it is sufficient to prove that any resource allocation mode j will have $\bar{v}_j^{(m)}(t) = 0$ if there is a k such that

$$(\nabla_m \bar{g}(t))_j > (\nabla_m \bar{g}(t))_k \quad (46)$$

which follows from the optimality shown in Lemma 3 along a convergent subsequence. For 2), we will prove that for any feasible points $\bar{\mathbf{u}}$ of (45), we have

$$\nabla_{\mathbf{z}} \bar{g}(t)^T \bar{\mathbf{u}}(t) \leq \nabla_{\mathbf{z}} \bar{g}(t)^T \bar{\mathbf{u}} \quad (47)$$

which also follows from the optimality in Lemma 3 along a convergent subsequence.

Proof: For the clarity of presentation, the proof is moved to the Appendix. ■

Based on the above lemma, we are now ready to prove that Algorithm 1 achieves the optimal cost in the fluid limit.

Lemma 7: For any fluid limit, we have for all $t > 0$,

$$g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t)) = g^* \quad (48)$$

where $g^* = f(\mathbf{x}_p^*; \mathbf{y}) + \beta p(\mathbf{x}_p^*, \mathbf{z}_p^*; \mathbf{y})$. Thus, the optimality is achieved in the fluid limit.

We first outline the proof. Note that it is always true that

$$g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t)) \geq g^* \quad (49)$$

since $(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t))$ are always feasible points of PEN. Thus, the claim holds if we can prove the reverse direction. This can be done by defining a proper ‘‘Lyapunov’’ function

$$L(t) = t g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t)) \quad (50)$$

and show that $\dot{L}(t) \leq g^*$, by using the properties in Lemma 6 and the convexity of function $g(\cdot)$.

Proof: Consider the ‘‘Lyapunov’’ function as in (50) in any fluid limit. From Lemma 5 we know that for any $t > 0$, $(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t))$ are feasible for PEN, and therefore we have $L(t) \geq g^* t$ due to the definition of g^* . On the other hand,

$$\begin{aligned} \dot{L}(t) &= \bar{g}(t) + t \dot{\bar{g}}(t) \\ &= \bar{g}(t) + t \nabla_{\mathbf{z}} \bar{g}(t)^T \dot{\bar{\mathbf{z}}}(t) + t \sum_{m \in \mathcal{M}} \nabla_m \bar{g}(t)^T \dot{\bar{\mathbf{x}}}^{(m)}(t) \\ &\stackrel{(a)}{=} \bar{g}(t) + t \nabla_{\mathbf{z}} \bar{g}(t)^T (\bar{\mathbf{u}}(t) - \bar{\mathbf{z}}(t)) \\ &\quad + \sum_{m \in \mathcal{M}} \nabla_m \bar{g}(t)^T (\bar{\mathbf{v}}^{(m)}(t) - \bar{\mathbf{x}}^{(m)}(t)) \\ &\stackrel{(b)}{\leq} \bar{g}(t) + t \nabla_{\mathbf{z}} \bar{g}(t)^T (\mathbf{z}_p^* - \bar{\mathbf{z}}(t)) \\ &\quad + \sum_{m \in \mathcal{M}} \nabla_m \bar{g}(t)^T (\mathbf{x}_p^{(m)*} - \bar{\mathbf{x}}^{(m)}(t)) \\ &\stackrel{(c)}{\leq} g(\mathbf{x}_p^*, \mathbf{z}_p^*) \stackrel{(d)}{=} g^* \end{aligned}$$

where (a) is obtained by substituting the equation in Lemma 5, (b) is from Lemma 6, i.e., $\bar{\mathbf{v}}^{(m)}(t)$ and $\bar{\mathbf{u}}(t)$ are solutions of GRAD-XBAR and GRAD-ZBAR, respectively. (c) is due to the convexity of function $g(\cdot)$, and (d) is because $(\mathbf{x}_p^*, \mathbf{z}_p^*)$ is the solution of PEN, by definition. Thus, we have

$$L(t) = L(0) + \int_0^t \dot{L}(\tau) d\tau \leq g^* t \quad (51)$$

from which we conclude that $L(t) = g^* t$. ■

Having established the optimality in the fluid limit, we are now able to prove optimality in the original system. The following theorem states that Algorithm 1 achieves the optimal cost in the original network.

Theorem 1: (optimal cost) In the original network, the following holds w.p.1:

$$\limsup_{t \rightarrow \infty} f(\mathbf{x}(t); \mathbf{y}(t)) \leq f_\epsilon^* \quad (52)$$

Proof: Suppose that it is not true. Then there is a sequence $\{t_n\}_{n=1}^\infty$ such that

$$\lim_{n \rightarrow \infty} f(\mathbf{x}(t_n); \mathbf{y}(t_n)) > f_\epsilon^* \quad (53)$$

From the Arzela-Ascoli Theorem [13], there is a subsequence $\{t_{n_k}\}_{k=1}^\infty$ which converges to a fluid limit. Lemma 7 implies

$$\lim_{k \rightarrow \infty} f(\mathbf{x}(t_{n_k})) \leq \lim_{k \rightarrow \infty} g(\mathbf{x}(t_{n_k})) \quad (54)$$

$$\stackrel{(a)}{=} g(\bar{\mathbf{x}}(1)) \quad (55)$$

$$\stackrel{(b)}{=} g^* \leq f_\epsilon^* \quad (56)$$

where (a) follows from the fact that for all $m \in \mathcal{M}$,

$$\mathbf{x}^{(m)}(t_{n_k}) = \mathbf{T}^{(m)}(t_{n_k}) / \mathbf{1}^T \mathbf{T}^{(m)}(t_{n_k}) \quad (57)$$

$$= (\mathbf{T}^{(m)})^{t_{n_k}}(1) / \mathbf{1}^T (\mathbf{T}^{(m)})^{t_{n_k}}(1) \quad (58)$$

$$\rightarrow \bar{\mathbf{T}}^{(m)}(1) / \mathbf{1}^T \bar{\mathbf{T}}^{(m)}(1) \text{ as } k \rightarrow \infty \quad (59)$$

$$= \bar{\mathbf{x}}^{(m)}(1) \quad (60)$$

and that $g(\cdot)$ is continuous. (b) is because of Lemma 7. Thus, we have a contradiction, and the claim holds. ■

In the next subsection we will continue to prove the feasibility result, namely, the limit points of $\mathbf{x}(t)$ produced by Algorithm 1 are indeed feasible for OPT.

C. Feasibility Proof

Note that Algorithm 1 is designed to solve PEN. Thus, in order to prove that the scheduler produce feasible points for OPT, we need the following lemma, which connects the objective function value in PEN to the constraint in OPT.

Lemma 8: The following properties hold for PEN: For large enough β , we have

$$\|\mathbf{h}(\mathbf{x}_p^*; \mathbf{y}) + \mathbf{z}_p^*\| \leq \epsilon/2 \quad (61)$$

for any solution $(\mathbf{x}_p^*, \mathbf{z}_p^*)$.

Proof: For the ease of presentation, we only consider the penalty function as (18), although the proof can be easily extended to general cases. Note that from Lemma 2 we have

$$\frac{\beta}{\alpha} \|\mathbf{h}(\mathbf{x}_p^*; \mathbf{y}) + \mathbf{z}_p^*\|^\alpha \leq f_\epsilon^* - f(\mathbf{x}_p^*; \mathbf{y}) \quad (62)$$

Thus, (61) holds by choosing sufficiently large β . ■

Finally, we conclude this section by the following theorem, which states that the limit points produced by Algorithm 1 are always feasible for the original problem OPT. This, combined with Theorem 1, proves the optimality of Algorithm 1 for OPT.

Theorem 2: (feasibility) For sufficiently large β , we have

$$\limsup_{t \rightarrow \infty} h_i(\mathbf{x}(t); \mathbf{y}(t)) \leq 0 \quad (63)$$

for any constraint function $h_i(\mathbf{x}; \mathbf{y})$ in $\mathbf{h}(\mathbf{x}; \mathbf{y})$.

Proof: Suppose that this is not true. Then there exist a sequence $\{t_n\}_{n=1}^{\infty}$ such that

$$\lim_{n \rightarrow \infty} h_i(\mathbf{x}(t_n); \mathbf{y}(t_n)) > 0 \quad (64)$$

From Arzela-Ascoli Theorem, there is a subsequence $\{t_{n_k}\}_{k=1}^{\infty}$ which converges to a fluid limit. Thus, we have

$$\begin{aligned} \|\mathbf{h}(\bar{\mathbf{x}}(1); \bar{\mathbf{y}}(1)) + \bar{\mathbf{z}}(1)\| &\geq h_i(\bar{\mathbf{x}}(1); \bar{\mathbf{y}}(1)) + \bar{z}_i(1) \\ &\stackrel{(a)}{=} \lim_{k \rightarrow \infty} h_i(\mathbf{x}(t_{n_k}); \mathbf{y}(t_{n_k})) + z_i(t_{n_k}) \\ &\stackrel{(b)}{\geq} \lim_{k \rightarrow \infty} h_i(\mathbf{x}(t_{n_k}); \mathbf{y}(t_{n_k})) + \epsilon \\ &\stackrel{(c)}{\geq} \epsilon \end{aligned}$$

where (a) can be argued similarly as in the proof of Theorem 1, (b) is because for any i and $t > 0$ we have $z_i(t) > \epsilon$, due to Algorithm 1, and (c) is because of the assumption in (64). But according to Lemma 7, $(\bar{\mathbf{x}}(t)), \bar{\mathbf{z}}(t)$ solves PEN, and therefore Lemma 8 implies that

$$\|\mathbf{h}(\bar{\mathbf{x}}(1); \bar{\mathbf{y}}(1)) + \bar{\mathbf{z}}(1)\| \leq \epsilon/2 \quad (65)$$

Contradiction! Therefore the claim holds. ■

Thus, Algorithm 1 produces feasible points for OPT, and achieves a cost which is arbitrarily close to f^* , by properly selecting parameters β and ϵ .

V. SIMULATION

In this section we verify the performance of Algorithm 1 through a simulation in a random wireless network where the network is as shown in Fig. 1. There are 7 links in the network, where square nodes denote the transmitters, and round nodes denote the receivers. We simulate a special case of OPT, the following minimum power scheduling problem, which we denote as POW:

$$\text{POW: } \min_{\mathbf{x}} \quad \mathbf{p}^T \mathbf{x} \quad (66)$$

$$\text{s.t.} \quad \mathbf{a} - G\mathbf{x} \preceq \mathbf{0} \quad (67)$$

$$\mathbf{x} \succeq 0, \quad \mathbf{1}^T \mathbf{x} = 1 \quad (68)$$

where \mathbf{p} is a power vector whose each element p_k corresponds to the power consumption when the independent set G_k is chosen. In the simulation we choose $p_k = \|G_k\|^2$. Here, \mathbf{a} corresponds to the arrival rate vector, which is assumed to be the only unknown parameter in the network. Thus, (67) corresponds to the rate stability constraint.

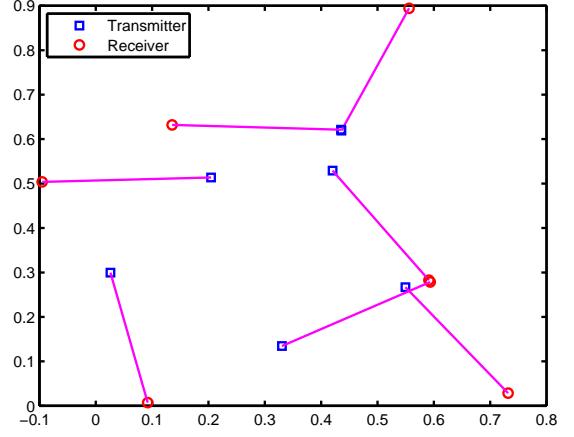


Fig. 1. An example wireless network with 7 links, where square nodes are transmitters, and round nodes are receivers.

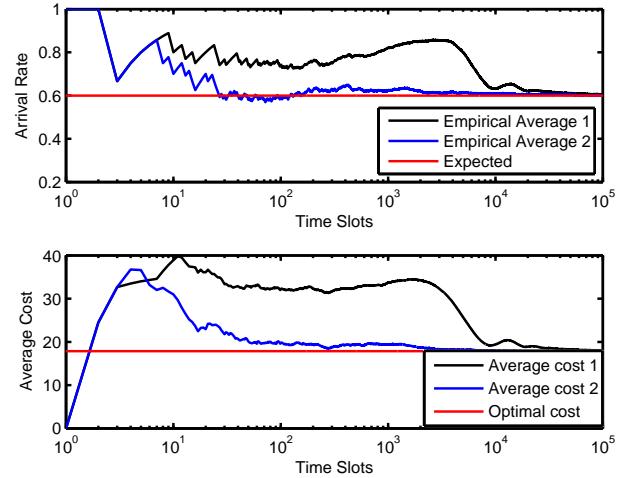


Fig. 2. Convergence results of average cost with sources with different convergence behaviors.

Fig. 2 shows the convergence results of the cost function (the bottom sub-figure) with slowly converging sources (the top sub-figure) after a simulation of 10^5 time slots. In the simulation, we choose $\epsilon = 10^{-3}$ and $\beta = 5 \times 10^3$. It can be observed from the top sub-figure that our algorithm achieves the optimal cost. Further, by comparing the convergence results of the cost and the arrival processes, we can conclude that Algorithm 1 can track the uncertain parameter \mathbf{a} dynamically. In the simulation, it is further observed that the maximum queue length in the network is around 10^2 , so that the constraint in (67) is clearly satisfied.

VI. CONCLUSION

In this paper we formulated a general class of scheduling problems in wireless networks with uncertain parameters, subject to the constraint that these parameters can be obtained from the empirical average values of certain stochastic network

processes. We proposed a class of primal-dual type scheduling algorithms, and showed its optimality as well as feasibility using fluid limits.

APPENDIX PROOF OF LEMMA 6

Proof: We first prove 1). Let a sequence of functions $(\mathbf{T}^{r_n}, \mathbf{Y}^{r_n}, \mathbf{Z}^{r_n})$ be given, which converge to a fluid limit $(\bar{\mathbf{T}}, \bar{\mathbf{Y}}, \bar{\mathbf{Z}})$. In the fluid limit, suppose that there is time $t > 0$, $m \in \mathcal{M}$ and resource allocation modes j, k such that

$$(\nabla_m \bar{g}(t))_j \geq (\nabla_m \bar{g}(t))_k + \epsilon \quad (69)$$

where $\epsilon > 0$ is a small constant. Then, since $\nabla_m \bar{g}(t)$ is a continuous function of variable t , there is $\delta_1 > 0$ such that for all $\tau \in (t - \delta_1, t + \delta_1)$, we have

$$(\nabla_m \bar{g}(\tau))_j \geq (\nabla_m \bar{g}(\tau))_k + \epsilon/2 \quad (70)$$

Further, since $\nabla_m \bar{g}(\tau) \triangleq \nabla_m g(\bar{\mathbf{x}}(\tau), \bar{\mathbf{z}}(\tau); \bar{\mathbf{y}}(\tau))$ is continuous as a function of variables $(\bar{\mathbf{x}}(\tau), \bar{\mathbf{y}}(\tau), \bar{\mathbf{z}}(\tau))$ (and therefore is absolutely continuous when restricted to a compact local region), there is an $\epsilon' > 0$ such that $\|(\mathbf{x}, \mathbf{y}, \mathbf{z}) - (\bar{\mathbf{x}}(\tau), \bar{\mathbf{y}}(\tau), \bar{\mathbf{z}}(\tau))\| \leq \epsilon'$ implies that

$$(\nabla_m g(\mathbf{x}, \mathbf{z}; \mathbf{y}))_j \geq (\nabla_m g(\bar{\mathbf{x}}(\tau), \bar{\mathbf{z}}(\tau); \bar{\mathbf{y}}(\tau)))_k + \epsilon/4$$

for all $\tau \in (t - \delta_1, t + \delta_1)$. Now we define

$$(\mathbf{x}^{r_n})^{(m)}(\tau) \triangleq \mathbf{T}^{(m)}(r_n \tau) / \mathbf{1}^T \mathbf{T}^{(m)}(r_n \tau) \quad (71)$$

$$\mathbf{y}^{r_n}(\tau) \triangleq \mathbf{Y}(r_n \tau) / r_n \tau \quad (72)$$

$$\mathbf{z}^{r_n}(\tau) \triangleq \mathbf{Z}(r_n \tau) / r_n \tau \quad (73)$$

Then, the definition of fluid limits implies that there exists $N \in \mathbb{N}$ and $\delta_2 > 0$ such that for all $n > N$ and $\tau \in (t - \delta_2, t + \delta_2)$,

$$\|(\mathbf{x}^{r_n}(\tau), \mathbf{y}^{r_n}(\tau), \mathbf{z}^{r_n}(\tau)) - (\bar{\mathbf{x}}(\tau), \bar{\mathbf{y}}(\tau), \bar{\mathbf{z}}(\tau))\| < \epsilon' \quad (74)$$

Thus, by taking $\delta = \min(\delta_1, \delta_2)$ we have

$$\begin{aligned} & (\nabla_m g(\mathbf{x}^{r_n}(\tau), \mathbf{z}^{r_n}(\tau); \mathbf{y}^{r_n}(\tau)))_j \\ & \geq (\nabla_m g(\mathbf{x}^{r_n}(\tau), \mathbf{z}^{r_n}(\tau); \mathbf{y}^{r_n}(\tau)))_k + \epsilon/4 \end{aligned}$$

for all $\tau \in (t - \delta, t + \delta)$. Further, by comparing the above definitions of $\mathbf{x}^{r_n}(\tau)$, $\mathbf{y}^{r_n}(\tau)$ and $\mathbf{z}^{r_n}(\tau)$ to that of $\mathbf{x}(\tau)$, $\mathbf{y}(\tau)$ and $\mathbf{z}(\tau)$ in (19), (21) and (22), respectively, we conclude that they are essentially the same, except a difference in time scale, i.e., $\mathbf{x}^{r_n}(t) = \mathbf{x}(r_n t)$. Thus, the following holds in the original system: for any $n > N$ and all $\tau \in (r_n(t - \delta), r_n(t + \delta))$,

$$(\nabla_m g(\mathbf{x}(\tau), \mathbf{z}(\tau); \mathbf{y}(\tau)))_j \geq (\nabla_m g(\mathbf{x}(\tau), \mathbf{z}(\tau); \mathbf{y}(\tau)))_k + \epsilon/4$$

Therefore, according to Lemma 3, $\xi_j^{(m)}$ is never chosen in any time slot during $(r_n(t - \delta), r_n(t + \delta))$, and we have that $T_j^{(m)}(\tau)$ is a constant during $(r_n(t - \delta), r_n(t + \delta))$, from which we conclude that $\dot{T}_j^{(m)}(t) = 0$. Therefore, $\bar{v}_j^{(m)}(t) = 0$ following the definition that $\bar{v}_j^{(m)}(t) = \dot{T}_j^{(m)}(t)/\pi^{(m)}$.

We next prove 2). Let $\bar{\mathbf{u}}$ be given as a feasible point of GRAD-ZBAR and $\epsilon > 0$ be given. Since $\nabla_z g(\cdot)$ is a

continuous function of t , there is $\delta > 0$ and $N \in \mathbb{N}$ such that for $n > N$ and all $\tau \in (t - \delta, t + \delta)$, the following holds:

$$\|\nabla_z g(\mathbf{x}^{r_n}(\tau), \mathbf{z}^{r_n}(\tau); \mathbf{y}^{r_n}(\tau)) - \nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))\| < \epsilon \quad (75)$$

Further, note that Lemma 3 implies that for any time slot in $(r_n(t - \delta), r_n(t + \delta))$, we have

$$\nabla_z g(\mathbf{x}(\tau), \mathbf{z}(\tau); \mathbf{y}(\tau))^T \bar{\mathbf{u}}(\tau) \leq \nabla_z g(\mathbf{x}(\tau), \mathbf{z}(\tau); \mathbf{y}(\tau))^T \bar{\mathbf{u}}$$

Thus, applying (75) to the above inequality we have

$$\nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))^T \bar{\mathbf{u}}(\tau) \leq \nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))^T \bar{\mathbf{u}} + c\epsilon$$

for all $\tau \in (r_n(t - \delta), r_n(t + \delta))$, where $c > 0$ is a proper constant. After summing over $(r_n(t - \delta), r_n(t + \delta))$ and dividing by r_n on both sides, we obtain

$$\begin{aligned} & \nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))^T (\mathbf{Z}^{r_n}(t + \delta) - \mathbf{Z}^{r_n}(t - \delta)) \\ & \leq 2\delta(\nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))^T \bar{\mathbf{u}} + c\epsilon) \end{aligned}$$

Finally, we let $n \rightarrow \infty$, and noting that $\delta > 0$ can be taken arbitrarily small, we have

$$\nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))^T \bar{\mathbf{u}}(t) \leq \nabla_z g(\bar{\mathbf{x}}(t), \bar{\mathbf{z}}(t); \bar{\mathbf{y}}(t))^T \bar{\mathbf{u}} + c\epsilon$$

from which 2) holds since $\epsilon > 0$ is arbitrary. ■

REFERENCES

- [1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. on Automatic Control*, Vol. 37, No. 12, pp. 1936-1949, December 1992
- [2] M. J. Neely, "Energy Optimal Control for Time Varying Wireless Networks", *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915-2934, July 2006
- [3] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Information Theory*, Vol. 34, Sept. 1988, pp. 910 - 917
- [4] L. Georgiadis, M. J. Neely, L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks," *Foundations and Trends in Networking*, Vol. 1, no. 1, pp. 1-144, 2006.
- [5] M. J. Neely, "Stochastic Network Optimization with Non-Convex Utilities and Costs," *Proc. Information Theory and Applications Workshop (ITA)*, Feb. 2010.
- [6] A.L. Stolyar, "Maximizing Queuing Network Utility subject to Stability: Greedy Primal-Dual Algorithm," *Queuing Systems*, 2005, Vol. 50, No.4, pp.401-457
- [7] A. Eryilmaz and R. Srikant, "Joint Congestion Control, Routing and MAC for Stability and Fairness in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, August 2006, 1514-1524.
- [8] A. Eryilmaz, and R. Srikant, "Fair Resource Allocation in Wireless Networks using Queue-length based Scheduling and Congestion Control", *IEEE/ACM Trans. on Networking*, Jan., 2007, pp. 1333-1344.
- [9] F. P. Kelly, A.K. Maulloo and D.K.H. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, Vol. 49 (1998), 237-252
- [10] S. Shakkottai and R. Srikant, "Network Optimization and Control," *Foundations and Trends in Networking*, Now Publishers, 2007.
- [11] D. P. Bertsekas, "Nonlinear Programming", *Athena Scientific*, 2nd Edition, 1999.
- [12] J.G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proc. of the IEEE INFOCOM*, 2:556-564, March 2000
- [13] H. Royden, "Real Analysis", *Prentice Hall*, 3rd Edition, 1988.